

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Комп'ютеризовані системи
управління»
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»
на тему: «Програмний застосунок для статистичного аналізу
пасажиропотоку громадського транспорту»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІА-61

Нагорний Михайло Костянтинович _____

Керівник:

К.т.н. доц. каф. АУТС,

Сокульський Олег Євгенович _____

Рецензент:

К.т.н., доц. каф. АСОІУ

Попенко Володимир Дмитрович _____

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютеризовані системи управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Нагорному Михайлу Костянтиновичу

1. Тема проєкту «Програмний застосунок для статистичного аналізу пасажиропотоку громадського транспорту», керівник проєкту Сокульський Олег Євгенович, к.т.н. доц. каф. АУТС, затверджені наказом по університету від « 7 » травня_ 2020 р. №1081-с

2. Термін подання студентом проєкту 11.06.2020

3. Вихідні дані до проєкту

C#, Visual Studio, SQL, хмарна технологія Azure.

4. Зміст пояснювальної записки

1. Вступ 2. Опис предметної області 3. Огляд існуючих рішень 4. Аналіз вимог до програмного забезпечення 5. Вибір технологій розробки 6. Розробка системи 7. Тестування програмного забезпечення 8. Впровадження та використання розробленої системи.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

Діаграма варіантів використання, діаграма компонентів, діаграма послідовності, діаграма структури бази даних.

6. Дата видачі завдання 13.04.2020

Календарний план

№ з / п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Вибір тематичного напрямку та узгодження теми дипломного проекту	22.02.2020	
2	Аналіз теоретичних матеріалів та вивчення предметної області	15.04.2020	
3	Розробка технічного завдання, вибір методів та засобів реалізації задачі	24.04.2020	
4	Огляд існуючих рішень з тематики роботи	27.04.2020	
5	Розробка структури прототипу та проектування системи	06.05.2020	
6	Реалізація проекту	20.05.2020	
7	Налагодження та перевірка програми	23.05.2020	
8	Оформлення пояснювальної записки	02.06.2020	
9	Передзахист дипломного проекту	03.06.2020	
10	Доопрацювання пояснювальної записки та підготовка презентації	13.06.2020	
11	Захист дипломного проекту	16.06.2020	

Студент

Михайло НАГОРНИЙ

Керівник

Олег СОКУЛЬСЬКИЙ

АНОТАЦІЯ

Нагорний М.К. Програмний застосунок для статистичного аналізу пасажиропотоку громадського транспорту. КПІ ім. Ігоря Сікорського, Київ, 2020.

Ключові слова: пасажиропотік, громадський транспорт, C#, Visual Studio, Windows Forms, SQL, хмарний сервіс Azure.

Основна частина документу викладена у пояснювальній записці, виконаній на 60 сторінках, та містить 1 рисунок та 27 таблиць. Також до його змісту входить 1 додаток.

Об'єктом розробки є програмний застосунок для статистичного аналізу пасажиропотоку громадського транспорту.

Мета розробки – створення програмного застосунку, що надає статистичну інформацію про пасажиропотік громадського транспорту та дає рекомендації по оптимізації розкладу руху транспортних засобів.

У дипломному проекті проведено ретельний аналіз та вибір технологій розробки програмного застосунку, які надають можливість швидкого та гнучкого налаштування. Створено програмний застосунок з використанням інтегрованого середовища розробки Visual Studio, мови програмування C# та мови програмування баз даних SQL.

Отримані результати можуть бути корисними при створенні аналогічних чи подібних систем.

SUMMARY

Nahorny M. Software application for statistical analysis of passenger traffic of public transport. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2020.

Key words: passenger traffic, public transport, C#, Visual Studio, Windows Forms, SQL, Azure cloud service.

The main part of the document consists of an explanatory note, which is made on 60 pages, and contains 1 figure and 27 tables. It also includes 1 appendix.

The object of development is a software application for statistical analysis of public transport traffic.

The purpose of the development is to create a software application that provides statistical information on the flow of public transport and provides recommendations for optimizing of the traffic timetable.

This diploma project carefully analyses and selects software application development technologies that provide quick and flexible configuration. A software application was developed using the integrated development environment Visual Studio, C# programming language and SQL database programming language.

The obtained results can be useful in creating similar systems.

Пояснювальна записка
до дипломного проєкту
на тему: «Програмний застосунок для
статистичного аналізу пасажиропотоку
громадського транспорту»

ЗМІСТ

ВСТУП.....	5
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Обґрунтування доцільності розробки	7
1.1.1 Опис та аналіз предметного середовища	7
1.1.2 Аналіз функціональних особливостей системи	8
1.2 Призначення розробки	10
1.3 Цілі та задачі розробки.....	10
1.4 Статистичний аналіз.....	11
1.4.1 Метод статистичного спостереження.....	11
1.4.2 Зведення і групування матеріалів статистичного спостереження	13
1.5 Висновок до розділу.....	15
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	16
2.1 Існуючі системи контролю пасажиропотоку	16
2.2 Висновок до розділу.....	18
3 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	19
3.1 Сценарії використання програмного застосунок.....	19
3.2 Розробка функціональні вимог до системи.....	22
3.3 Розробка нефункціональних вимог до системи	23
3.4 Висновок до розділу.....	23
4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ	24
4.1 Переваги обраних технологій розробки	24
4.1.1 Мова програмування C#.....	25
4.1.2 Visual Studio	26
4.1.3 Windows Forms.....	27
4.1.4 Конструктор Windows Forms	28

					<i>IA61.160BAK.005 ПЗ</i>		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмний застосунок для статистичного аналізу пасажиропотоку громадського транспорту</i>		
<i>Розроб.</i>		<i>Нагорний М.К.</i>					
<i>Перевір.</i>		<i>Сокульський О.Є.</i>					
<i>Реценз.</i>							
<i>Н. Контр.</i>							
<i>Затверд.</i>							
					<i>Лім.</i>	<i>Арк.</i>	<i>Акрушіє</i>
						2	60
					<i>КПІ ім. Ігоря Сікорського ФІОТ, гр. ІА-61</i>		

4.1.5 Мова програмування баз данихSQL	28
4.1.6 Система управління базами даних PostgreSQL.....	30
4.1.7 Графічний клієнт для управління базами даних pgAdmin	31
4.1.8 Редактор баз даних в Visual Studio	31
4.1.9 Система контролю версій GitHub	32
4.1.10 Хмарна технологія Azure	33
4.2 Висновок до розділу.....	33
5 РОЗРОБКА СИСТЕМИ	34
5.1 Опис програми.....	34
5.1.1 Опис інтерфейсу	34
5.1.2 Програмний опис	36
5.2 Опис роботи системи	40
5.3 Розробка бази даних.....	42
5.4 Висновок до розділу.....	45
6 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	46
6.1 Фази життєвого циклу програмного забезпечення.....	46
6.2 Тестування програмного забезпечення	47
6.5 Висновок до розділу.....	54
7 ВПРОВАДЖЕННЯ ТА ВИКОРИСТАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ.....	55
7.1 Апаратні та програмні вимоги для експлуатації програми.....	55
7.2 Інструкція користувача	56
7.3 Висновок до розділу.....	57
ВИСНОВКИ	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	59

ВСТУП

На сьогоднішній день маршрутні таксі забезпечують біля 25% загального пасажирського трафіку Києва та покривають усе місто маршрутами різної протяжності.

Далеко не всі маршрути є оптимальними з точки зору логістики пасажироперевезень, що породжує незручність в першу чергу для пасажирів та в деяких випадках призводить до ускладнення їх маршрутів. Кількість автобусів на кожному з маршрутів, відповідно й частота їх прибуття, в ідеалі повинні залежати від затребуваності маршруту, тобто від того, яка кількість людей має потребу використовувати його регулярно. Але найчастіше ця оцінка проводиться поверхнево і багато в чому залежить від суб'єктивних вражень водіїв. Крім того, щільність графіку руху автобусів не завжди залежить від того, наскільки сильно вони наповнені в певний проміжок часу: часто виникають ситуації, коли автобуси їдять майже порожні або навпаки, переповнені.

Для вирішення цієї проблеми диспетчери маршрутів повинні мати сучасний робочий інструмент для обробки статистичних даних щодо пасажиропотоку маршрутних таксі на всіх маршрутах. Маючи ці дані в реальному масштабі часу вже можна гнучко реагувати на зміни у пасажиропотоці та корегувати графік ходження автобусів на маршрутах.

Перш за все диспетчеру необхідний зручний засіб відображення в масштабі реального часу зібраних статистичних даних з інформативним та інтуїтивно зрозумілим інтерфейсом, так, щоб користувачу було одразу зрозуміло, які з маршрутів переповнені чи навпаки, недостатньо наповнені, на яких ділянках та в який час доби.

Для збору статистики буде використана спеціальна камера, яка за допомогою програми, написаною у Raspberry Pi, рахуватиме усіх людей, що заходять та виходять з транспортного засобу, скануючи їх обличчя та відсилаючи отриману інформацію до розташованої у хмарі бази даних. Програма, розроблена

					IA61.160БАК.005 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

у ході виконання дипломної роботи буде працювати з даними саме з цієї бази даних. Ця програма в зручному і зрозумілому вигляді відображує статистичні дані щодо маршрута, а також здатна проводити поверхневий аналіз оптимальності розкладу руху автобусів на маршруті. Передбачено використання даних GPS, які будуть передаватися з камер, встановлених в автобусах.

На даний момент не існує сервісів, що можуть цілеспрямовано та об'єктивно оцінювати завантаженість громадського транспорту. Існують додатки, що дозволяють опосередковано визначати завантаженість роботи пасажирського транспорту в тому числі і маршрутних таксі. Перш за все це Google Maps, що аналізує загальний стан руху на дорогах. Проте цей сервіс задовольняє перш за все потреби автомобілістів, оскільки оцінює дорожній трафік. Тим не менш використання даних, які надає цей сервіс, для побудови або зміни маршрутів можливе. Також існують сервіси, що за допомогою супутників відстежують положення певних транспортних засобів. У великих містах більшість пасажироперевезень, що здійснюються комунальними засобами на міських маршрутах, постійно відстежуються, та будь-хто з пасажирів може за допомогою смартфона завчасно дізнатись, коли до них прибуде очікуваний транспорт.

Проте жоден з цих сервісів не дає змоги збирати статистику для оптимізації розкладу руху якого-небудь з видів громадського транспорту. Саме тому був розроблений програмний комплекс, що буде збирати статистику та зводити поточну інформацію, подавати її в зручному форматі та робити поверхневий аналіз цих даних. Це дозволить людям, що займаються розробкою розкладу руху маршрутного таксі, розробляти нові поліпшені маршрутні розклади та оптимізувати існуючі.

					IA61.160БАК.005 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Обґрунтування доцільності розробки

Вітчизняних систем збору статистики та аналізу пасажиропотоку міського громадського транспорту на даний момент не існує. Фактично розрахунок кількості пасажирів ведеться лише на основі обліку кількості оплачених поїздок. Проте цей спосіб не є надійним та не надає повну інформацію.

По-перше, підрахунок кількості оплачених поїздок у громадському транспорті не враховує тих людей, що проїзд не оплачують.

По-друге, якщо вести підрахунок лише кількості оплачених поїздок, то можливо дізнатись лише загальну кількість перевезених пасажирів за певний період часу. Інформацію про те, скільки людей входить та виходить на тих чи інших зупинках, коли саме, на яких ділянках маршруту та напрямках руху відбувається найбільше навантаження, отримати за рахунок обліку надходжень неможливо ніяким чином.

Система збору статистики та статистичного аналізу пасажиропотоку позбавлена цих недоліків, адже фактично враховує усіх пасажирів.

1.1.1 Опис та аналіз предметного середовища

Програмний застосунок для статистичного аналізу пасажиропотоку громадського транспорту — це програмне забезпечення, головною функцією якого є надання диспетчеру статистичних даних щодо пасажиропотоку громадського транспорту, що відслідковується системою збору статистики, а також надання загальних рекомендацій для оптимізації розкладу руху транспорту.

Зазвичай, такі системи будуються на основі клієнт-серверної архітектури[1]. Клієнт-серверна архітектура складається з розподілених застосунків і передбачає взаємодію та обмін даними між ними.

					IA61.160BAK.005 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Складовими частинами клієнт-серверної архітектури є такі компоненти:

- сервери, які надають дані клієнтським програмам, що її запитують;
- клієнти, які використовують дані, що надають сервери;
- мережа, що надає можливість взаємодії між серверами і клієнтами.

Використання програмного застосунку для статистичного аналізу пасажиропотоку громадського транспорту слід розглянути в першу чергу зі сторони користувача.

Користувачем програмного застосунку є оператор, який розробляє маршрути для громадського транспорту та складає розклад руху. Розроблений в ході роботи над дипломним проектом програмний застосунок полегшить роботу оператору, надасть можливість системно аналізувати одержувані дані, що підвищить якість його роботи. Інформативний та інтуїтивно зрозумілий інтерфейс програми надасть всю необхідну інформацію, що стосується аналізуемого користувачем маршруту, а також програма може дати загальні рекомендації щодо оптимізації маршруту, звернувши увагу на те, що деякі ділянки маршруту в певний час перевантажені або навпаки, занадто мало навантажені. Ця інформація допоможе більш ефективно складати розклад руху громадського транспорту, даючи змогу розподіляти автобуси, а також інші транспортні засоби, по тим маршрутам, на яких є перевантаження, при цьому забираючи транспортні засоби з тих, на яких їх і так забагато.

1.1.2 Аналіз функціональних особливостей системи

Програмний застосунок для статистичного аналізу пасажиропотоку громадського транспорту являє собою самостійний додаток, функціонал якого дозволяє дізнатися загальну інформацію щодо певного маршруту та отримати загальні рекомендації по оптимізації розкладу руху транспортних засобів на маршруті. Підтримується функціонал авторизації користувачів та реєстрації нових.

					IA61.160БАК.005 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Складається програмний застосунок для статистичного аналізу пасажиропотоку громадського транспорту з декількох основних компонентів:

- елементи вводу;
- елементи виводу;
- система взаємодії з базою даних;
- система рекомендацій по оптимізації.

Для взаємодії програмного застосунку та користувача присутні елементи вводу. Елементи вводу представлені текстовими полями та кнопками, що при використанні їх користувачем дають змогу впливати на програмний застосунок, в першу чергу на елементи виводу. Текстові поля слугують для вводу значень, інформація щодо яких потрібна користувачеві. Кнопки слугують для того, щоб безпосередньо оновлювати дані, виведені в програмі, на ті, що стосуються даних, які користувач наново ввів у текстові поля.

Для виведення інформації у програмному застосунку присутні елементи виводу. Елементи виводу представлені таблицями, діаграмами та полем, в якому знаходиться зображення маршруту на мапі міста. Значення, які виводяться в елементах виводу, розраховуються програмно та безпосередньо не змінюються користувачем. Їх зміна можлива непрямым шляхом завдяки елементам вводу.

Система взаємодії з базою даних є прихованою від користувача. Завдяки ній програма має можливість виводити інформацію, що стосуються маршрутів, дані про які присутні в базі даних. База даних знаходиться в хмарі [2] та взаємодіє з програмою завдяки мережі Інтернет.

Система рекомендації по оптимізації працює лише після натискання відповідної кнопки. Її головною функцією є виведення загальних рекомендацій по оптимізації розкладу руху громадського транспорту. Розрахунок ведеться за програмним шляхом без впливу на нього зі сторони користувача. Користувач має можливість лише обрати маршрут, рекомендації щодо розкладу руху на якому необхідно вивести.

					IA61.160БАК.005 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Призначення розробки

Розроблений застосунок призначений для диспетчерів-операторів транспортних компаній, які розробляють маршрути громадського транспорту та складають розклад руху транспортних засобів на цьому маршруті. Програмний застосунок полегшує роботу оператора та підвищує її ефективність, допомагаючи скласти більш оптимальний розклад.

Програмний застосунок є новим видом програмного забезпечення, аналогів якому на даний момент ще не створювалось. Тому його призначенням, окрім безпосередньої мети його розробки, є створення нової ринкової ніші серед програмного забезпечення. Сам програмний застосунок є достатньо гнучким та легко розширюваним, тобто при необхідності існує можливість зміни вже існуючого функціоналу та додавання нового. Можливості розробки можуть бути суттєво підвищені за умов її інтеграції з комунальними комп'ютерними системами. Крім того створення затребуваного програмного застосунку у новій ринковій ніші, що поки не має аналогів на ринку, має привести до появи великої кількості нових програм від різних розробників.

1.3 Цілі та задачі розробки

Цілі розробки цього застосунку можна умовно розділити на дві категорії: для розробників і для користувачів системи.

Головна ціль створення цього застосунку для розробників – це створення першої самостійної системи для аналізу статистичних даних щодо пасажиропотоку громадського транспорту, можливість подальшого вдосконалення існуючої системи та створення нових систем на основі отриманого досвіду.

Головна ціль створення цього застосунку для користувачів системи – спрощення роботи з розроблення маршрутів громадського транспорту та

					IA61.160БАК.005 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

складання більш оптимальних розкладів руху транспортних засобів на цих маршрутах.

1.4 Статистичний аналіз

Статистичне дослідження [3] може проводитися за допомогою наступних методик:

- статистичне спостереження;
- зведення і групування матеріалів статистичного спостереження;
- абсолютні і відносні статистичні величини;
- варіаційні ряди;
- вибірка;
- кореляційний і регресійний аналіз;
- ряди динаміки.

В ході роботи над дипломним проектом було використано методи статистичного спостереження та зведення і групування матеріалів статистичного спостереження. Дані методи були обрані тому що дають інформацію у найбільш простому та при цьому достатньо розгорнутому вигляді. Що стосується об'єкту дослідження, то в даному випадку це пасажиропотік міського громадського транспорту.

1.4.1 Метод статистичного спостереження

Статистичне спостереження є планомірним, організованим і в більшості випадків систематичним збором інформації, спрямованим, головним чином, на явища соціального життя. Реалізується даний метод через реєстрацію попередньо визначених найбільш вагомих ознак. Її метою є подальше встановлення характеристик досліджуваних явищ.

					IA61.160БАК.005 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Статистичне спостереження повинне виконуватися з урахуванням деяких важливих вимог:

- воно повинно повністю охоплювати досліджувані явища;
- одержані дані повинні бути точними і достовірними;
- одержані дані повинні бути одноманітними і легко порівнюваними.

Також статистичне спостереження може мати дві форми:

– Звітність – форма статистичного спостереження, що передбачає заповнення спеціально затверджених документів (звітів), за якими відповідні статистичні підрозділи організацій, установ або підприємств отримують від респондентів необхідну інформацію про їх діяльність. Статистичну звітність подають з підписами осіб, які відповідають за достовірність наданої інформації.

– Спеціально організоване спостереження – форма статистичного спостереження, що охоплює сфери суспільного життя та діяльності, які не відображаються у звітності, або ж для уточнення і встановлення достовірності інформації звітів.

Метод статистичного спостереження використовується в першу чергу при зборі інформації стосовно пасажиропотоку міського громадського транспорту та наповнення бази даних. За допомогою камер та запрограмованої плати Raspberry Pi проводиться підрахунок пасажирів, що знаходяться у транспортному засобі, за яким ведеться спостереження. Автоматична система обробки даних про кількість пасажирів в транспортних засобах фіксує зупинку транспортного засобу, за допомогою модуля GPS визначає своє місцезнаходження, а після того, як транспортний засіб зрушує, проводить підрахунок кількості пасажирів за допомогою функції розпізнавання осіб. Після цього отримані дані відсилаються на хмару та зберігаються у базі даних, що розташована у ній.

					IA61.160БАК.005 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

1.4.2 Зведення і групування матеріалів статистичного спостереження

Зведення представляє собою процес обробки певних одиничних фактів, які утворюють загальну сукупність даних, зібраних під час спостереження. Якщо зведення проводиться грамотно, величезна кількість одиничних даних про окремі об'єкти спостереження може перетворитися в цілий комплекс статистичних таблиць і результатів. Також таке дослідження сприяє визначенню спільних рис і закономірностей досліджуваних явищ.

З урахуванням показників точності і глибини вивчення можна виділити просте і складне зведення, але будь-яке з них має ґрунтуватися на конкретних етапах:

- вибирається ознака групування;
- визначається порядок формування груп;
- розробляється система показників, що дозволяють охарактеризувати групу і об'єкт або явище в цілому;
- розробляються макети таблиць, де будуть представлені результати зведення.

Є різні форми зведення:

- централізоване зведення, що вимагає передачі отриманого первинного матеріалу до вищестоящого центру для подальшої обробки;
- децентралізоване зведення, де вивчення даних відбувається на декількох рівнях по висхідній.

Виконуватися же зведення може за допомогою спеціалізованого обладнання, наприклад, з використанням комп'ютерного ПО або вручну.

Що ж стосується угруповання, то цей процес відрізняється поділом досліджуваних даних на групи за ознаками. Особливості поставлених статистичним аналізом завдань впливають на те, яким саме буде угруповання: типологічним, структурним або аналітичним.

					ІА61.160БАК.005 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Метод зведення і групування матеріалів статистичного спостереження покладений у основу алгоритмів, що використовуються вже безпосередньо програмним застосунком. Автоматична система обробки даних про кількість пасажирів в транспортних засобах лише збирає дані стосовно пасажиропотоку та відсилає їх до бази даних, без розподілу по категоріям. Програмний застосунок, у свою чергу, приймає дані з бази та виводить їх у тому вигляді, в якому вони будуть корисними для користувача. Проводиться первинне зведення даних по категоріям. По-перше, це виведення загальної інформації, що стосується певного маршруту, інформації щодо транспортних засобів, задіяних на цьому маршруті, кількості пасажирів, які були перевезені за певний проміжок часу, та розподілу пасажирів на певних зупинках в певний час. По-друге, це виведення більш деталізованої інформації за бажанням користувача. А саме: розподіл пасажирів на певних зупинках в певний час в обмежених проміжках часу та завантаженість застосованих транспортних засобів на певних ділянках маршруту.

Крім того, цей метод статистичного аналізу використовується у вбудованій в програмний застосунок системі надання рекомендацій стосовно оптимізації розкладу руху транспортних засобів, що використовуються на обраному маршруті. Ця система групує дані по зупинкам на три категорії: занадто велике завантаження транспортного засобу, оптимальне завантаження транспортного засобу та занадто мале завантаження транспортного засобу. Вже на основі цих груп проводиться розрахунок оптимізації розкладу руху транспортних засобів. На ті маршрути, де в певний час на постійній основі завантаженість транспортних засобів не є оптимальною, вводиться спеціальний коефіцієнт збільшення чи навпаки зменшення кількості транспортних засобів, задіяних на маршруті, або заміни рухомого складу на маршруті на одиниці більшої, чи навпаки, меншої місткості.

					IA61.160БАК.005 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

1.5 Висновок до розділу

Розглянувши існуючі проблеми зі збором та аналізом статистики використання пасажирями громадського транспорту, а також шляхів вирішення цих проблем, можна зробити декілька висновків.

По-перше, сучасні методи розрахунку пасажиропотоку не є оптимальними і не дають всієї необхідної інформації щодо того, коли саме та на яких ділянках маршруту завантаженість транспортних засобів є занадто низькою, високою чи оптимальною.

По-друге, при розробці програмного застосунку необхідно щоб він давав користувачеві достатню кількість структурованої інформації щодо пасажиропотоку на обраному маршруті, а його інтерфейс був простим, зручним та інтуїтивно зрозумілим.

По-третє, робота операторів транспортних компаній, що розробляють маршрути громадського транспорту та складають розклад його руху, може бути полегшена, а якість результату їх роботи підвищена.

По-четверте, застосування методів статистичного аналізу в якості важливої частини програмного застосунку дозволяє значно розширити функціонал кінцевого продукту, покращуючи можливості його взаємодії з користувачем та даючи змогу створювати на основі цих методів нові програмні блоки.

Крім того, слід зазначити, що створення унікального програмного застосунку, що не має безпосереднього аналогу, може стати поштовхом для створення нової ніші серед програмних додатків.

					IA61.160БАК.005 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

2.1 Існуючі системи контролю пасажиропотоку

На даний момент збір статистики з використання громадського транспорту ведеться, в першу чергу, за допомогою підрахунку кількості людей, що сплачують за проїзд. Використання технічних засобів, таких як турнікети в метро та спеціальні валідатори [4], що їх почали встановлювати в багатьох комунальних засобах громадського транспорту, наприклад в тролейбусах, автобусах та трамваях, значно підвищує точність зібраної статистичної інформації. Проте ці електронні засоби використовуються ще не всіма пасажирами. Дехто досі використовує паперові квитки, що продаються кондукторами, чи проходять у метро за посвідченнями, підтверджуючими пільги, без необхідності проходити через автоматичні турнікети.

В маршрутних же таксі, де немає реально працюючої системи білетів, підрахунок пасажирів взагалі ведеться через вирахування їх кількості з загального об'єму виручки, що була задекларована водіями маршруток. Точність такого підрахунку є сумнівною.

Жоден з цих засобів на даний момент не є надійним. До речі не враховуються ще й ті пасажирів, що їздять «зайцем», тобто без оплати проїзду. Наприклад, з січня 2018 по січень 2019 офіційною статистикою було зафіксовано зниження кількості пасажирів в громадському транспорті більш ніж на 10%, і частково це пояснюється тим, що через збільшення цін на проїзд кількість людей, які не платили в цей період за проїзд, значно зросла.

Система збору та аналізу статистики з використання громадського транспорту, розроблена в ході реалізації дипломного проекту, буде позбавлена цих недоліків. Кількість пасажирів, що знаходяться у транспорті, буде підраховуватись технічними засобами незалежно від того, чи сплачували вони за проїзд.

					IA61.160БАК.005 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Крім того, існуючі електронні системи оплати проїзду навіть за бездоганного функціонування не дають можливості детальної і своєчасної реєстрації завантаженості транспортного засобу в певний момент часу. Навіть за умови скасування паперових проїзних документів часто виникають ситуації коли оплата проїзду проходить не в момент посадки, а значно пізніше, а головне – ці системи оплати не фіксують момент, коли пасажир покидає салон транспортного засобу. У тих випадках, коли підрахунок кількості пасажирів ведеться без використання електронних засобів, тобто розрахунком через об'єм проданих квитків чи просто сплачених грошей, оцінити наскільки саме велике навантаження на певних відрізках маршруту та в певний час, неможливо в принципі.

Розроблена в ході реалізації дипломного проекту система дає можливість позбавитись цього недоліку, адже є наведеною саме на те, щоб реєструвати завантаженість не загальну, а в певний момент часу. Тим не менш загальна так само фіксується.

Сьогодні у всіх навігаційних сервісів головним засобом отримання інформації про завантаженість шляхів є зворотній зв'язок користувачів за допомогою GSM-зв'язку [5]. Люди їдуть по дорогах – смартфон або навігатор з GSM-модулем, на якому працює навігаційний сервіс, знає місце розташування і швидкість автомобіля і автоматично відправляє ці дані на сервер з певною періодичністю. Дані аналізуються, і на їх основі складається швидкісна карта, де інтенсивність руху відзначається по візуально зрозумілим принципом: зелений – їдемо, жовтий – помірна завантаженість, червоний – стоїмо.

Такі системи використовуються компанією Google та іншими для того, щоб слідити за завантаженістю на дорогах. Вони створені для безпосереднього аналізу трафіку, а не для інформування скільки людей, коли та де користуються громадським транспортом. Тим не менш їх використання для побудови маршрутів може бути корисним, завдяки визначенню напрямків та завантаженості основних міських пасажиропотоків.

					IA61.160БАК.005 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Розглянуті засоби збору інформації щодо пасажиропотоку громадського транспорту не здатні видати всієї необхідної інформації щодо завантаження транспортних засобів. Було виявлено декілька основних проблем:

- не враховуються пасажирів, які не оплатили проїзд;
- не враховується, коли та де пасажир сів до транспортного засобу;
- не враховується, коли та де пасажир покинув транспортний засіб.

Саме для вирішення цих проблем і створений програмний застосунок.

2.2 Висновок до розділу

Для того, щоб створити свій програмний продукт, необхідно детально вивчити існуючі рішення і виділити у них сильні та слабкі сторони. Це необхідно для того, щоб на основі цих даних розробити продукт, який перевершує розглянуті аналоги, не має недоліків, притаманних ним та якнайбільше відповідає саме нашим умовам.

В даному випадку, коли не існує аналогів серед інших програмний застосунків, вивчення існуючих систем, що здатні хоча б частково замінити створений, є навіть більш важливим, оскільки вони можуть задати необхідний напрям розробки. Цей напрям має відрізнятися від вже існуючих систем, проте доповнюватися розробленими в них рішеннями.

Зараз для побудови розкладу руху транспорту використовуються лише припущення про те, в який час навантаження на маршрут найбільше і саме в цей час збільшується кількість транспортних засобів, тобто реальна завантаженість транспортних засобів не враховується.

А системи підрахунку кількості пасажирів взагалі не відрізняються точністю та правдивістю своїх даних.

					IA61.160БАК.005 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

3 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Сценарії використання програмного застосунку

На кресленику ІА61.160БАК.005 Д1 наведена діаграма варіантів використання (діаграма прецедентів [6]).

Усі варіанти використання системи представлені у таблиці 3.1.

Таблиця 3.1 – Варіанти використання системи

Шифр варіанту використання	Назва варіанту використання
UC-1	Перегляд повної інформації щодо маршруту.
UC-1.1	Введення номеру маршруту.
UC-1.2	Натискання кнопки «Оновлення».
UC-2	Перегляд часової статистики обраного маршруту.
UC-2.1	Введення часового проміжку.
UC-2.2	Натискання кнопки «Часова характеристика».
UC-3	Перегляд статистики за зупинками обраного маршруту.
UC-3.1	Введення проміжку маршруту.
UC-3.2	Натискання кнопки «Характеристика за зупинками».
UC-4	Натискання кнопки «Рекомендації».
UC-5	Авторизація користувача.
UC-5.1	Реєстрація користувача.

Опис всіх варіантів використання програмного застосунку:

– УС-1 Перегляд повної інформації щодо маршруту. Користувач має змогу переглядати повну інформацію щодо маршруту на елементах виведення. Варіант використання відбувається при початку роботи програмного застосунку, коли виводиться інформація про маршрут за замовчанням, чи при виборі користувачем нового маршруту для перегляду. В цей момент на елементах виведення застосунку виводиться загальна інформація, що стосується обраного маршруту.

– УС-1.1 Введення номеру маршруту. Користувач може вводити необхідний номер маршруту в текстове поле «Номер маршруту». При введенні номеру у випадяючому переліку виводяться можливі значення номерів маршрутів, що відповідають даним, які вводяться.

– УС-1.2 Натискання кнопки «Оновлення». Користувач натискає кнопку «Оновлення». При виконанні варіанту використання всі дані, виведені в елементах виведення, змінюються на ті, що стосуються обраного в текстовому полі «Номер маршруту». Якщо текстове поле порожнє чи введені в нього дані не відповідають тим, що присутні в базі, то елементи виведення стають порожніми та виводиться повідомлення про помилку.

– УС-2 Перегляд часової статистики обраного маршруту. Користувач має змогу переглядати часову статистику обраного маршруту на діаграмі «Часова характеристика». Варіант використання виконується при запуску програмного застосунку чи при виборі користувачем нового номеру маршруту.

– УС-2.1 Введення часового проміжку. Користувач може вводити необхідний йому часовий проміжок в текстові поля «Часова характеристика». В ліве текстове поле вводиться значення початку часового проміжку, в праве – значення кінця проміжку.

– УС-2.2 Натискання кнопки «Часова характеристика». Користувач натискає кнопку «Часова характеристика». При виконання варіанту використання дані, що виводяться на діаграмі «Часова характеристика», змінюються на ті, що стосуються введеного в текстові поля «Часова

					ІА61.160БАК.005 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

характеристика» проміжку часу. Якщо дані в текстових полях були введені з помилкою, то виводиться повідомлення про помилку.

– УС-3 Перегляд статистики за зупинками обраного маршруту. Користувач має змогу переглядати статистику за зупинками обраного маршруту на діаграмі «Характеристика за зупинками». Варіант використання виконується при запуску програмного застосунку чи при виборі користувачем нового номеру маршруту.

– УС-3.1 Введення проміжку маршруту. Користувач може вводити необхідний проміжок маршруту в текстові поля «Характеристика за зупинками». В ліве текстове поле вводиться значення першої зупинки з необхідного проміжку, в праве – значення останньої зупинки проміжку.

– УС-3.2 Натискання кнопки «Характеристика за зупинками». Користувач натискає кнопку «Характеристика за зупинками». При виконанні варіанту використання дані, що виводяться на діаграмі «Характеристика за зупинками», змінюються на ті, що стосуються обраного проміжку маршруту. Якщо дані в текстових полях були введені з помилкою, то виводиться повідомлення про помилку.

– УС-4 Натискання кнопки «Рекомендації». Користувач натискає кнопку «Рекомендації». При виконанні варіанту використання відкривається вікно рекомендацій, в якому виводяться загальні рекомендації по оптимізації розкладу руху транспорту на маршруті, номер якого введено в текстове поле «Номер маршруту»

– УС-5 Авторизація користувача. Користувач має змогу авторизуватися в системі. Варіант використання виконується при натисканні кнопки «Авторизація», відкритті вікна авторизація, введенні користувачем персональних даних та натисканні кнопка «Авторизація» у вікні авторизації. Після виконання варіанту використання вікно авторизація закривається, а користувач авторизується. Якщо персональні дані, введені користувачем, не

відповідають даним, що містяться у базі даних, замість авторизація та закриття вікна виводиться повідомлення про помилку.

– UC-5.1 Реєстрація користувача. Користувач має змогу зареєструватися в системі. Варіант використання виконується при натисканні користувачем кнопки «Авторизація», відкритті вікна авторизації, натисканні користувачем кнопки «Реєстрація», відкритті вікна реєстрація, введення користувачем особистих даних для реєстрації та натискання кнопки «Реєстрація» у вікні реєстрації. Після виконання варіанту використання вікно реєстрація закривається, а новий користувач реєструється. Якщо якісь з введених користувачем особистих даних співпадають з тими, що знаходяться в базі, то замість реєстрації та закриття вікна реєстрації виводиться повідомлення про помилку.

3.2 Розробка функціональні вимог до системи

Функціональні вимоги до системи є відповідними щодо варіантів використання:

- перегляд повної інформації щодо маршруту;
- введення номеру маршруту;
- натискання кнопки «Оновлення»;
- перегляд часової статистики обраного маршруту;
- введення часового проміжку;
- натискання кнопки «Часова характеристика»;
- перегляд статистики за зупинками обраного маршруту;
- введення проміжку маршруту;
- натискання кнопки «Характеристика за зупинками»;
- натискання кнопки «Рекомендації»;
- авторизація користувача;
- реєстрація користувача.

					IA61.160БАК.005 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3 Розробка нефункціональних вимог до системи

Проаналізувавши предметну область та здійснивши огляд існуючих рішень, можна скласти наступні нефункціональні вимоги до програмного забезпечення:

- розширюваність – програмний застосунок має бути достатньо гнучким для того, щоб у майбутньому була можливість розширювати функціонал за допомогою створення додаткових модулів;
- переносимість – програмний застосунок має підтримувати перенесення та розгортання його на різноманітних платформах;
- повторне використання компонентів – компоненти програмного застосунку мають бути розроблені таким чином, щоб у майбутньому була можливість їх повторного використання;
- можливість автоматизування тестування – програмний застосунок повинен мати інструментарій для проведення автоматизованого тестування;
- підтримка сучасними операційними системами – увесь функціонал програмного застосунку має підтримуватись у всіх версіях операційної системи Windows.

3.4 Висновок до розділу

У даному розділі на базі інформації, що була зібрана та проаналізована у попередніх розділах, було створено діаграму варіантів використання, а також були висунуті функціональні та нефункціональні вимоги до програмного застосунку. Варто зазначити, що правильно складені вимоги слугують базою для частин майбутнього програмного застосунку, чим полегшують подальшу розробку та тестування програмного забезпечення.

					IA61.160БАК.005 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

4.1 Переваги обраних технологій розробки

При створенні програмних продуктів потрібно дослідити існуючі технології і практики їх використання для того, щоб обрати для розробки найкращі та найсучасніші з них.

Зазвичай розробка будь-якого програмного застосунку вимагає використання декількох технологій розробки. Наприклад, для створення самостійного програмного застосунку часто використовують мови програмування такі як C# та Java, а для серверної частини спеціальну мову для розробки баз даних, наприклад SQL.

Вибір технології залежить від багатьох факторів пов'язаних як із самою системою, так і з командою розробки.

Перший фактор, який впливає на вибір технології – стек навичок команди розробки. У більшості випадків кожен розробник має технології, середовища розробки, мови програмування або розмітки, яким він надає перевагу, з якими він найчастіше працює та має певні напрацювання.

Ще одним фактором при виборі технології є особливості і вимоги до програмного застосунку, що розробляється. Такими факторами можуть бути навантаження на систему, швидкість виконання, сумісність технології з апаратною частиною тощо.

Також варто зазначити, що деякі технології можуть базуватися на інших, використовувати їх або бути сумісними з ними.

При розробці цього програмного застосунку були використані наступні технології:

- мова програмування C#;
- середовище розробки Visual Studio;
- інтерфейс програмування Windows Forms;
- вбудований в Visual Studio редактор форм;

					IA61.160БАК.005 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

- мова програмування баз даних SQL;
- система управління базами даних PostgreSQL;
- графічний клієнт для управління базами даних pgAdmin;
- вбудований в Visual Studio редактор баз даних;
- система контролю версій GitHub
- хмарний сервіс для зберігання баз даних Azure.

4.1.1 Мова програмування C#

C# - об'єктно-орієнтована мова програмування [7].

На сьогоднішній момент мова програмування C# швидко розвиваються і є однією з найпотужніших і затребуваних мов в ІТ-галузі. На даний момент на ній пишуться найрізноманітніші програми: від невеликих десктопних програмок до великих веб-порталів і веб-сервісів, які обслуговують щодня мільйони користувачів.

C# є мовою з Сі-подібним синтаксисом і близька в цьому відношенні до C++ і Java.

C# є об'єктно-орієнтованою і в цьому плані багато перейняла у Java і C++. Наприклад, C# підтримує поліморфізм, успадкування, перевантаження операторів, статичну типізацію. Об'єктно-орієнтований підхід дозволяє вирішити завдання з побудови великих, але в той же час гнучких, масштабованих і розширюваних додатків.

Мова програмування C# дуже тісно пов'язана с платформою .NET. Основними перевагами платформи є:

- підтримка декількох мов. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує кілька мов: поряд з C# це також VB.NET, C++, F#, а також різні діалекти інших мов, прив'язані до .NET, наприклад, Delphi. NET;
- кросплатформеність. .NET є переносимою платформою;

					IA61.160БАК.005 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

– потужна бібліотека класів. .NET представляє єдину для всіх підтримуваних мов бібліотеку класів;

– різноманітність технологій. Загальномовне середовище виконання CLR і базова бібліотека класів є основою для цілого стека технологій, які розробники можуть задіяти при побудові тих чи інших додатків.

Мова програмування C# була обрана для написання програмного застосунку оскільки універсальність та потужність цієї мови дозволяє реалізувати всі необхідні програмні компоненти.

4.1.2 Visual Studio

Інтегроване середовище розробки Visual Studio - це стартовий майданчик для написання, налагодження і складання коду, а також подальшої публікації додатків [8]. Інтегроване середовище розробки (IDE) являє собою багатофункціональну програму, яку можна використовувати для різних аспектів розробки програмного забезпечення. Крім стандартного редактора і відладчика, які існують в більшості середовищ IDE, Visual Studio включає в себе компілятори, засоби автозавершення коду, графічні конструктори і багато інших функцій для спрощення процесу розробки.

Ефективні засоби підвищення ефективності розробки, доступні у Visual Studio:

– хвилясті лінії і швидкі дії. Хвилясті лінії позначають помилки або потенційні проблеми коду прямо під час введення. Якщо навести курсор миші на хвилясту лінію, на екран будуть виведені додаткові відомості про помилку. Крім того, в поле зліва може з'являтися значок лампочки зі швидкими діями щодо усунення помилки;

– очистка коду. Очищення коду допомагає усунути багато проблем в коді ще до перевірки коду;

					IA61.160БАК.005 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

- рефакторинг. Рефакторинг включає в себе такі операції, як інтелектуальне перейменування змінних, витягування однієї або декількох рядків коду в новий метод, зміна порядку параметрів методів і багато іншого;
- IntelliSense. IntelliSense - це набір функцій, що відображають відомості про код безпосередньо в редакторі і в деяких випадках автоматично створюють невеликі уривки коду;
- поле пошуку. Поле пошуку дозволяє швидко знайти потрібний вміст в Visual Studio;
- Live Share. Надає можливість спільного редагування та налагодження в реальному часі незалежно від типу додатка або мови програмування;
- ієрархія викликів. У вікні Ієрархія викликів показані методи, що викликають обраний метод;
- CodeLens. CodeLens допомагає знаходити посилання на код, зміни коду, пов'язані помилки, робочі елементи, перевірки коду та модульні тести.

Написання програмного застосунку виконується за допомогою засобів програмування середовища розробки Visual Studio оскільки це середовище програмування дає всі необхідні засоби для роботи з мовою програмування C#.

4.1.3 Windows Forms

Windows Forms - це інтелектуальна клієнтська технологія для .NET Framework, набору керованих бібліотек, які спрощують поширені завдання програми, такі як читання та запис у файлову систему [9]. При використанні такого середовища розробки, як Visual Studio, є можливість створювання смарт-клієнтських програм Windows Forms, що відображають інформацію, вимагають введення даних від користувачів та спілкуються з віддаленими комп'ютерами по мережі.

У Windows Forms форма - це візуальна поверхня, на якій відображається інформація для користувача. Зазвичай при створенні програми Windows Forms,

					IA61.160БАК.005 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

додаються елементи керування у форми та розробляються відповіді на дії користувача, такі як натискання миші або натискання клавіш. Керування - це елемент дискретного користувацького інтерфейсу (UI), який відображає дані або приймає дані.

4.1.4 Конструктор Windows Forms

Конструктор Windows Forms надає багаточисельні засоби для створення додатків Windows Forms [10]. Він надає такі можливості для створення форм:

- розміщення елементів управління за допомогою ліній прив'язки;
- виконання завдання конструктора за допомогою смарт-тегів;
- встановлення поля і відбиття для елементів управління;
- розташування елементів управління за допомогою елемента управління `TableLayoutPanel`;
- розподіл макета елемента управління за допомогою елемента управління `SplitContainer`;
- перегляд макета у вікні «Структура документа»;
- розміщення елементів управління з відображенням розміру і відомостей про розташування;
- встановлювати значення властивостей у вікні «Властивості».

Технологія Windows Forms була обрана для написання програмного застосунку оскільки дозволяє максимально швидко створювати інтерфейси програм та налаштовувати їх.

4.1.5 Мова програмування баз данихSQL

SQL (Structured Query Language - Структурована мова запитів) - мова управління базами даних для реляційних баз даних [11]. Сама по собі SQL не є Тьюринг-повною мовою програмування, але її стандарт дозволяє створювати для

					IA61.160БАК.005 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

неї процедурні розширення, які розширюють її функціональність до повноцінної мови програмування.

SQL складається з чотирьох окремих частин:

- мова визначення даних (DDL) використовується для визначення структур даних, що зберігаються в базі даних. Оператори DDL дозволяють створювати, редагувати і видаляти окремі об'єкти в БД. Допустимі типи об'єктів залежать від використовуваної СУБД і зазвичай включають бази даних, користувачів, таблиці і ряд дрібніших допоміжних об'єктів, наприклад, ролі і індекси;

- мова маніпуляції даними (DML) використовується для вилучення і зміни даних в БД. Оператори DML дозволяють витягати, вставляти, змінювати і видаляти дані в таблицях. Іноді оператори select вилучення даних не розглядаються як частина DML, оскільки вони не змінюють стан даних. Всі оператори DML носять декларативний характер;

- мова визначення доступу до даних (DCL) використовується для контролю доступу до даних в БД. Оператори DCL застосовуються до привілеїв і дозволяють видавати і відбирати права на застосування певних операторів DDL і DML до певних об'єктів БД;

- мова управління транзакціями (TCL) використовується для контролю обробки транзакцій в БД. Зазвичай оператори TCL включають commit для підтвердження змін, зроблених в ході транзакції, rollback для їх скасування та savepoint для розбиття транзакції на кілька менших частин.

Мова програмування баз даних SQL була обрана для написання бази даних для програмного застосунку адже вона є найпростішою для виконання цієї задачі та надає всі необхідні можливості.

4.1.6 Система управління базами даних PostgreSQL

PostgreSQL - це вільно поширювана об'єктно-реляційна система управління базами даних найбільш розвинена з відкритих СУБД в світі і є реальною альтернативою комерційним базам даних [12].

Основні переваги PostgreSQL:

- надійність, яка забезпечується завдяки відповідності принципам ACID, багатоверсійності, наявності Write Ahead Logging, можливості відновити базу даних, репліційності та відкритості кодів;
- продуктивність, яка ґрунтується на використанні індексів, інтелектуальному планувальнику запитів, тонкої системи блокувань, системі управління буферами пам'яті і кешування, чудовою масштабованістю при конкурентній роботі;
- можливість розширення, яка означає, що користувач може налаштовувати систему шляхом визначення нових функцій, агрегатів, типів, мов, індексів і операторів. Об'єктно-орієнтованість PostgreSQL дозволяє перенести логіку додатка на рівень бази даних, що сильно спрощує розробку клієнтів, так як вся бізнес логіка знаходиться в базі даних. Функції в PostgreSQL однозначно визначаються назвою, кількістю і типами аргументів.
- підтримка SQL;
- багатий набір типів даних PostgreSQL;
- простота використання.

PostgreSQL був обраний в якості системи управління базами даних оскільки він являє собою потужний безкоштовний засіб, що дозволяє створювати бази даних та працювати з ними. Крім того, ця система легко з'єднується з програмами, написаними завдяки .Net.

					IA61.160БАК.005 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

4.1.7 Графічний клієнт для управління базами даних pgAdmin

pgAdmin являє собою графічний клієнт для роботи з сервером, через який в зручному вигляді можливо створювати, видаляти, змінювати бази даних і управляти ними[13].

Протягом багатьох років pgAdmin є стандартним вільно доступним клієнтським графічним клієнтом для PostgreSQL і поставляється у багатьох встановлених пакетах. Він пропонує інструмент запитів SQL, редактор процедурних мов та інтерфейс CRUD. Це також один з небагатьох клієнтів, який надає графічний інтерфейс GUI для налагоджувача plpgsql.

Безкоштовний центр адміністрування бази даних PostgreSQL. Включає графічний інтерфейс адміністрування, інструмент запитів SQL, редактор процедурного коду та багато іншого. Графічний інтерфейс підтримує всі функції PostgreSQL та полегшує адміністрування.

pgAdmin був обраний через свою простоту при використанні, достатньо широкий вибір інструментів для роботи з базами даних та відкритість доступу до нього.

4.1.8 Редактор баз даних в Visual Studio

Існує кілька способів створення баз даних в SQL Server [14]. За допомогою набору інструментів SQL Enterprise Manager бази даних можна створювати графічно або програмно (за допомогою команд на мові SQL). Крім нього, існують багаточисельні зовнішні інструменти для створення баз даних, наприклад Visio, який буде описано далі.

Visual Studio .NET також містить дуже зручний інструмент для роботи з базами даних SQL Server. Він входить до складу нового компонента Server Explorer, який призначений для централізованого управління всіма видами серверного програмного забезпечення.

					IA61.160БАК.005 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

Категорії об'єктів баз даних, з якими працює вбудований в Visual Studio редактор:

- діаграми баз даних. ;
- таблиці;
- представлення;
- збережені процедури;
- функції.

Вбудований в Visual Studio редактор баз даних використовується для того, щоб приєднати побудовану базу даних до програмного застосунку, а також для приєднання до хмари.

4.1.9 Система контролю версій GitHub

GitHub – це web-сервіс для зберігання проектів, що використовує систему контролю версій Git [15]. Зазвичай використовується командами розробників для сумісної роботи над проектами.

Особливості:

- GitHub має вбудовану систему відстеження проблем та помилок;
- даний сервіс має функцію графічного відображення гілок у проекті;
- GitHub надає можливість для сумісної роботи декількох або навіть великої кількості розробників;
- сервіс має функціонал, що дозволяє розробникам з усього світу пропонувати свої виправлення та покращення коду;

У даному проекті GitHub було використано для збереження коду проекту на серверах сервісу.

					IA61.160БАК.005 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

4.1.10 Хмарна технологія Azure

Microsoft Azure - це набір хмарних служб, що постійно розширюється, призначений допомагати організаціям вирішувати бізнес-завдання [16]. Сервіс дозволяє створювати, розгортати додатки і управляти ними в великій глобальній мережі з використанням різних інструментів і платформ.

Переваги Azure:

- постійні оновлення. Майкрософт безперервно впроваджує інновації;
- різноплановість. Завдяки підтримці всіх додатків і платформ і можливості використовувати відкритий код існує можливість створювати додатки зручним способом і розгортати їх в потрібних середовищах;
- проста робота з гібридною хмарою;
- надійність;
- безкоштовність.

Хмарна технологія Azure була обрана через доступність, адже дає можливість використовувати хмарне сховище без необхідності сплачувати.

4.2 Висновок до розділу

У цьому розділі на підставі аналізу предметної області, огляду існуючих рішень, вимог до розроблюваної системи та особистого досвіду було обрано технології, які були використані при розробці програмного застосунку для статистичного аналізу пасажиропотоку громадського транспорту.

Використання різних технологій при розробці програмного застосунку є обов'язковим, оскільки не існує універсальних технологій, які були б здатні до створення програмного забезпечення будь-якого виду, яке складалось би з декількох різних пов'язаних між собою модулів.

					IA61.160БАК.005 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

5 РОЗРОБКА СИСТЕМИ

5.1 Опис програми

5.1.1 Опис інтерфейсу

На рисунку 5.1 зображено інтерфейс програмного застосунку.

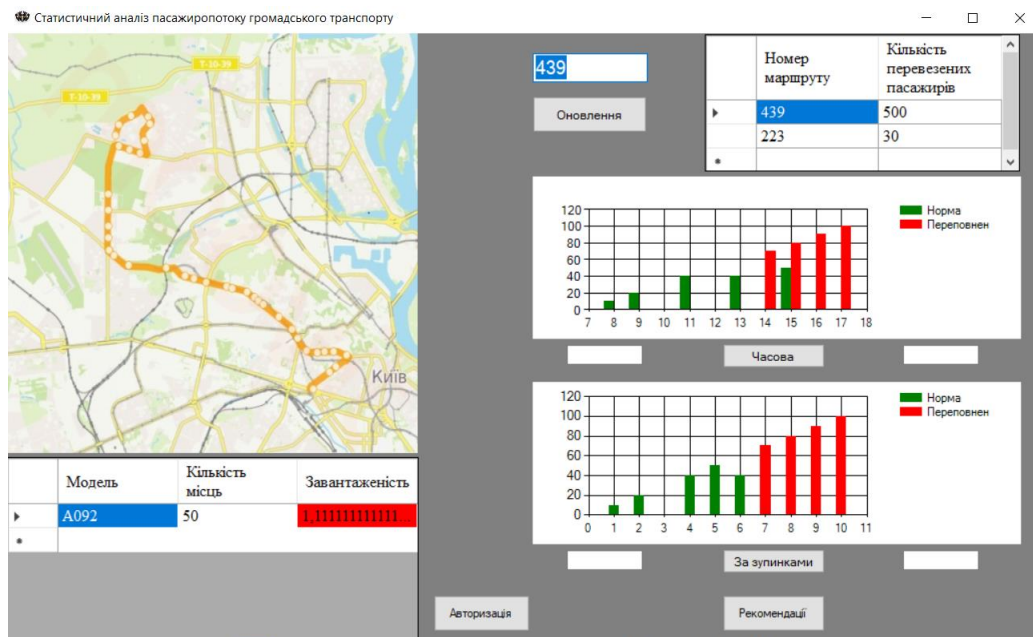


Рисунок 5.1 – Головна форма програмного застосунку

Інтерфейс програмного застосунку складається з наступних елементів:

1. Текстове поле «Номер маршруту». Приймає значення номера маршруту, інформацію по якому необхідно переглянути.
2. Кнопка «Оновлення». При натисканні скидаються значення текстових полів «Часова характеристика» та «Характеристика за зупинками», виводиться оновлена інформація стосовно маршруту, введенного в поле «Номер Маршруту».
3. Поле «Карта міста». Відображає схему обраного в полі «Номер Маршруту» маршруту з позначеними номерами зупинок на карті міста.
4. Динамічна таблиця «Завантаженість». Виводиться інформація щодо коефіцієнта навантаження транспорту на маршруті, відповідно до задіяних моделей автобусів і їх базових характеристик місткості. Поле «Коефіцієнт навантаження» має інтуїтивно зрозумілу кольорову індикацію: якщо коефіцієнт

навантаження нижче або дорівнює одиниці, то клітина буде зеленою, а якщо вище одиниці, то червоною.

5. Статична таблиця «Загальна характеристика». Відображає загальну статистику щодо денної кількості пасажирів перевезених на кожному з маршрутів, що наявні в базі даних. Дані змінюються при оновленні бази даних.

6. Динамічна діаграма «Часова характеристика». На ній виводиться завантаженість автобусів маршруту, номер якого введений в текстове поле “Номер маршруту”, на ділянці, яку визначає користувач. Ділянка визначається вводом відповідних значень в текстові поля «Часова характеристика». Діаграма показує завантаженість ділянки маршруту з погодинним розподілом усього денного часу роботи маршруту.

7. Текстові поля «Часова характеристика». В них вводяться номери зупинок на маршруті, між якими необхідно вивести інформацію на діаграмі «Часова характеристика».

8. Кнопка «Часова характеристика». При натисканні оновлює діаграму «Часова характеристика» відповідно до змін, внесених в текстові поля «Часова характеристика».

9. Динамічна діаграма «Характеристика за зупинками». На ній виводиться завантаженість автобусів маршруту, номер якого введений в текстове поле «Номер маршруту», в певний проміжок часу, вказаний користувачем. Проміжок часу визначається вводом відповідних значень в текстові поля «Характеристика за зупинками». Діаграма показує завантаженість автобусів маршруту на усіх зупинках на обраному проміжку часу.

10. Текстові поля «Характеристика за зупинками». В них вводяться певні години, які утворюють проміжок часу, щодо якого виводиться інформація на діаграмі “Характеристика за зупинками”.

11. Кнопка «Характеристика за зупинками». При натисканні оновлює діаграму «Характеристика за зупинками» відповідно до змін в текстових полях «Характеристика за зупинками».

					ІА61.160БАК.005 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

12. Кнопка «Рекомендації». При натисканні відкриває нове вікно, в якому виводяться рекомендації по оптимізації руху транспорту на обраному маршруті відповідно до розрахунків за даними, що наявні в базі.

13. Кнопка «Авторизація». При натисканні відкриває нове вікно, в якому користувач має змогу авторизуватися в системі.

5.1.2 Програмний опис

Програма написана мовою програмування C# за допомогою середовища програмування Visual Studio 2019. Проект побудований на Windows Forms. В проекті використовується ООП.

При створенні об'єктів використовувалися абстрагування та інкапсуляція.

Проект містить три класи: route, bus та stop. Об'єкти саме цих класів створюються при прийнятті інформації з бази даних. Вони містять основну інформацію, що необхідна для роботи програми.

Об'єкти класу bus представляють собою окремі автобуси, що курсують на маршрутах, статистика по яким збирається. Об'єкт містить базову інформацію по цим автобусам: їх модель, номер маршруту, за яким вони курсують, та їх оптимальна місткість, яка дорівнює кількості сидячих місць, збільшену на 5.

Об'єкти класу stop представляють собою окремі зупинки, що їх роблять усі автобуси, статистика по яким збирається. Об'єкт містить всю необхідну інформацію щодо зупинок: автобус, який зробив зупинку, номер зупинки, що відповідає її номеру на карті, час, коли відбулася зупинка, кількість людей, що залишились в автобусі на момент покидання автобусом зупинки.

Об'єкти класу route представляють собою окремі маршрути, що є занесеними до бази даних. Об'єкт містить всю необхідну інформацію щодо маршрутів: номер маршруту, кількість зупинок та адреси цих зупинок. До них додатково прив'язуються зображення, на яких присутнє зображення мапи міста із відображенням на ній цього маршруту, з додатковими помітками на місцях

					ІА61.160БАК.005 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

зупинок. Проте цього зображення немає в базі і його необхідно прикріпляти до колекції елемента ілюстрацій окремо.

В програмі використовуються декілька патернів [17] для спрощення та оптимізації роботи:

1. Адаптер (Adapter) призначений для перетворення інтерфейсу одного класу в інтерфейс іншого. Використовується для перенесення інформації з бази даних. За допомогою нього дані з бази перетворюються в об'єкти програми.

2. Будівельник (Builder) - шаблон проектування, який інкапсулює створення об'єкта і дозволяє розділити його на різні етапи. Використовується для того, щоб незалежні один від одного об'єкти могли створюватися самостійно.

3. Патерн Ітератор (Iterator) надає абстрактний інтерфейс для послідовного доступу до всіх елементів складеного об'єкта без розкриття його внутрішньої структури. Використовується для того, щоб отримувати інформацію щодо всіх об'єктів, що стосуються обраного користувачем маршруту. При цьому доступу до зв'язку між об'єктами користувач не має.

4. Декоратор (Decorator) представляє структурний шаблон проектування, який дозволяє динамічно підключати до об'єкта додаткову функціональність. Використовується для можливості зміни об'єктів в ході розробки, якщо з'явиться така необхідність. Наприклад, при зміні бази даних.

Головна форма програми була створена завдяки вбудованому в Visual Studio редактору форм та містить декілька елементів:

Windows Forms текстові поля використовуються для отримання вхідних даних від користувача або для виведення тексту. Елемент управління TextBox зазвичай використовується для тексту, що редагується, хоча його також можна зробити доступним тільки для читання. Текстові поля можуть відображати кілька рядків, переносити текст в розмір елемента управління і додавати базове форматування. Елемент управління TextBox надає один стиль формату для тексту, що відображається або вводяться в елемент управління.

					IA61.160БАК.005 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

TextBox використовуються в елементах текстове поле «Номер маршруту», текстових полях «Часова характеристика» і текстових полях «Характеристика за маршрутом». В полі «Номер маршруту» використовується функція автозаповнення поля. Ця функція реалізована завдяки властивості AutoCompleteCustomSource, що дозволяє завдяки додаванню до її колекції списку наявних номерів маршрутів виводити їх у випадяючому списку при натисканні на текстове поле, а при початку введення номера кількість елементів у випадяючому списку зменшується, залишаючи лише ті, частини номерів яких було введено.

Елемент управління Windows Forms PictureBox використовується для виведення зображень в форматі точкових малюнків, GIF, JPEG, метафайлів і значків.

PictureBox використовується в елементі поле «Карта міста». До колекції додаються зображення маршрутів з позначеними на них номерами зупинок. За допомогою властивості Image виводяться зображення, що стосуються певного номера маршруту.

Елемент управління DataGridView надає потужний і гнучкий спосіб відображення даних в табличному форматі. Елемент управління DataGridView можна використовувати для відображення уявлень невеликого обсягу даних тільки для читання, або можна масштабувати його для відображення редагованого уявлення дуже великих наборів даних. Зовнішній вигляд елемента управління DataGridView легко налаштовується завданням значень декількох властивостей. Як джерело даних можуть використовуватися сховища даних різних типів, також елемент управління DataGridView може працювати без прив'язаного до нього джерела даних.

DataGridView використовується в елементах таблиця «Завантаженість» та таблиця «Загальна статистика». Значення в клітини таблиці вносяться програмно. Можливість редагування вимкнена завдяки властивості ReadOnly, значення якої переведено в true. Проте це не забороняє користувачеві, при

					IA61.160БАК.005 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

необхідності, виділяти та копіювати текст. Клітинки останнього стовпчика таблиці «Завантаженість» мають властивість `BackColor`, завдяки якому клітинки, в залежності від значення, зафарбовуються зеленим або червоним.

Елемент управління `Chart` - це об'єкт діаграми, що надає події. При додаванні діаграми на лист `Visual Studio` створює об'єкт `Chart`, який можна запрограмувати безпосередньо.

`Chart` використовується в елементах діаграма «Часова характеристика» та діаграма «Характеристика за зупинками». За допомогою властивості `Series` до елементів управління додається колекція значень, за якими буде побудована діаграма. Колекції додаються через редактор форм `Visual Studio` у властивостях колекції значень. Значення же, які додаються до колекції, додаються програмним шляхом. Графічні елементи діаграм також зафарбовуються у зелений або червоний колір завдяки властивості `Color`, яка стосується кожної окремої колекції. Саме колекцій у кожному елементі по дві. В першу колекцію додається інформація щодо тих зупинок, завантаженість на яких менше оптимальної. В другу же додається інформація про перевантажені зупинки.

Елемент управління `Button`. Найбільш часто використовуваним елементом управління є кнопка. Обробляючи подія натискання кнопки, ми може виробляти ті чи інші дії. При натисканні на кнопку на формі в редакторі `Visual Studio` ми за замовчуванням потрапляємо в код обробника події `Click`, який буде виконуватися при натисканні.

Кнопки використовуються в елементах кнопка «Оновлення», кнопка «Часова характеристика», кнопка «Характеристика за зупинками» та кнопка «Рекомендації». До кожної кнопки прив'язана подія `ButtonClick`, яка означає виконання певних дії при натисканні кнопки. Події кнопок «Оновлення», «Часова характеристика» та «Характеристика за зупинками» оновлюють значення, виведені на елементах виведення програми. Кнопка «Рекомендації» відкриває нову форму в окремому вікні, на якій виводяться загальні рекомендації по оптимізації обраного в текстовому полі «Номер маршруту» маршруту.

					ІА61.160БАК.005 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Модуль рекомендації, що відповідає за роботу елементів виведення у вікні рекомендації, працює за рахунок спеціального алгоритму оптимізації. Алгоритм складається з декількох етапів:

1. Збір та зведення інформації. При відкритті вікна оптимізації модуль оптимізації збирає всю існуючу інформацію, яка стосується обраного в текстовому полі «Номер маршруту» маршруту. Перш за все це перелік зупинок, які були зроблені транспортними засобами на цьому маршруті, а також інформація про ці зупинки. Ця інформація ділиться на три категорії: зупинки, на яких транспортний засіб було перевантажено, зупинки, на яких транспортний засіб був оптимально завантажений та зупинки, на яких завантаженість транспортних засобів була значно менша за оптимальну. Зупинки з другої категорії модулем не використовуються.

2. Аналіз інформації. Проводиться аналіз зупинок з перевантаженими транспортними засобами та зі слабо завантаженими транспортними засобами. Перш за все прибираються ті зупинки, на яких відхилення від оптимальної кількості пасажирів не являється частим. Потім прибираються ті, на яких часто присутнє відхилення, проте на сміжних зупинках такого відхилення не спостерігається. І вже потім обираються відрізки маршруту, на яких часто в один і той же час спостерігається відхилення від оптимальної кількості пасажирів транспортних засобів.

3. Побудова рекомендацій. Обираються ділянки маршруту, на яких присутні регулярні відхилення від оптимальної завантаженості транспортних засобів. Розраховується коефіцієнт, на який необхідно помножити чи поділити кількість транспортних засобів, задіяних в цей час на маршруті, аби оптимізувати кількість пасажирів у транспортних засобах. Крім того, якщо на маршруті задіяні транспортні засоби із занадто великою чи навпаки занадто низькою місткістю, то алгоритм може порекомендувати задіяти інші транспортні засоби, місткість яких краще відповідає завантаженості на маршруті.

					ІА61.160БАК.005 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

5.2 Опис роботи системи

На кресленику ІА61.160БАК.005 Д2 наведено діаграму компонентів.

З діаграми видно, що розроблений програмний застосунок має клієнт-серверну архітектуру: клієнтом виступає пристрій користувача, а саме – програмний застосунок, встановлений на ньому. Серверна частина складається з трьох основних компонентів: хмарного серверу Azure, бази даних та системи керування базами даних SQLite.

Діаграма послідовності, наведена на кресленику ІА61.160БАК.005 Д3, відображає принцип взаємодії програмного застосунку з базою даних, який можна відобразити на прикладі послідовності виконуваних дій при вході користувача у систему:

- спочатку користувач в програмному застосунку виконує дію – відкриває вікно авторизації, вводить дані для входу та натискає кнопку «Авторизація»;
- програмний застосунок надсилає запит з даними користувача на хмару;
- хмарний сервіс за допомогою системи керування базами даних відсилає запит до бази даних;
- база даних, розташована у хмарі, проводить пошук даних, які співпадають з запитом;
- після цього в базі даних формується відповідь, яка містить або дозвіл на авторизацію, або помилку, і передає її на хмарний сервіс;
- хмарний сервіс надсилає згенеровану відповідь до програмного застосунку;
- застосунок завершує процедуру авторизації.

					ІА61.160БАК.005 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

5.3 Розробка бази даних

Діаграму структури бази даних наведено на кресленику ІА61.160БАК.005 Д4, а опис і призначення таблиць бази даних зазначено у таблиці 5.1.

Таблиця 5.1 – Призначення таблиць бази даних.

Назва таблиці	Опис
stop	Таблиця зберігає дані про зупинки транспортних засобів. Містить у собі: ідентифікатор зупинки (id), ідентифікатор транспортного засобу, що зробив зупинку (vehicle_id), кількість пасажирів, що знаходилось у транспортному засобі на момент покидання зупинки (passenger_number), тривалість зупинки (timestamp) та локацію зупинки (location).
vehicle	Таблиця зберігає загальні дані про транспортні засоби, за якими ведеться спостереження. Містить у собі: ідентифікатор транспортного засобу (id), назва моделі транспортного засобу (vehicle_model), кількість сидячих місць у транспортному засобі (sits_number) та номер маршруту, на якому ходить транспортний засіб (route_number).
route	Таблиця зберігає дані по заданим зупинкам на маршруті, транспортні засоби на яких відстежуються.

Продовження таблиці 5.1

route	Містить у собі: ідентифікатор маршруту (id), номер маршруту (route_number), номер зупинки (order_number) та локацію зупинки (location)
user	Таблиця, що містить у собі дані про користувачів. Напрямку не пов'язана з іншими. Містить у собі: ідентифікатор користувача (id), логін користувача (user_login) та пароль користувача (user_password).
user_role	З'єднуюча таблиця, яка об'єднує таблиці user та role. Містить у собі: ідентифікатор користувача (user_id) та ідентифікатор ролі користувача (role_id).
role	Таблиця, що містить у собі дані по ролям користувачів. Містить у собі: ідентифікатор ролі (role_id) та назву цієї ролі (role_name).

Для взаємодії програмного застосунку та розташованої в хмарі бази даних використовується спеціальна бібліотека Npgsql.

Програмний застосунок підключається до бази даних після введення даних, що дають змогу підключитися до певної бази даних, розташованої у хмарі. А саме це адрес серверу, безпосередньо назва бази даних, до якої необхідно підключитися, номер порту підключення, ім'я аккаунта, яким саме буде проведено підключення, та пароль цього аккаунта.

Після цього створюється екземпляр класу NpgsqlConnection. Це спеціальний клас з бібліотеки Npgsql, який представляє підключення до бази даних. Його використання та задання певних даних необхідно для підключення

					IA61.160БАК.005 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

до сервера бази даних PostgreSQL. Перш за все екземпляру цього класу задається дія Open, яка безпосередньо і встановлює з'єднання програмного застосунку та бази даних.

Після встановлення з'єднання з'являється можливість відправляти до бази даних у хмарі спеціальних операторів, які дозволяють працювати з базою. В даному проекті були використані стандартні оператори мови програмування баз даних SQL.

Для формування команд використовуються екземпляри класу NpgsqlCommand, до властивостей яких якраз у вигляді строки задається безпосередньо оператор. При створення екземплярів цього класу задається, куди саме будуть записані дані, які будуть отримані від бази. Якщо до бази даних було подано запит на додавання нової інформації чи зміну структури бази, то повернена інформація представляє собою лише інформацію, яка підтверджує виконання команди. Якщо запит був на отримання інформації з бази, то повертається безпосередньо інформація.

В програмному застосунку не використовувались оператори, які змінюють структуру бази даних. Використовувались лише оператори для запису інформації та прийняття інформації з бази даних.

Оператор SELECT здійснює вибірку з бази даних і має найбільш складну структуру серед усіх операторів мови SQL.

Оператор SELECT використовується в програмному застосунку для прийняття інформації з бази даних та створення на основі цієї інформації об'єктів, з якими вже працює програма. Сама програма не містить в собі бази даних, тому при запуску програми та при оновленні даних в програмі вона кожного разу звертається до бази та приймає з неї інформацію. При цьому старі об'єкти стираються, а створюються нові. Деякі з запитів за допомогою оператора SELECT нічого не записують до програми нових даних, а лише перевіряють наявність у базі необхідних даних. Це стосується функцій авторизації та реєстрації.

					IA61.160БАК.005 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Оператор INSERT вставляє нові записи в таблицю. При цьому значення стовпців можуть являти собою літеральні константи, або бути результатом виконання підзапиту. У першому випадку для вставки кожного рядка використовується окремий оператор INSERT; у другому випадку буде вставлено стільки рядків, скільки повертається підзапитом.

Оператор INSERT використовується в програмному застосунку лише у функції реєстрації нового користувача. При цьому спочатку відсилається запит за допомогою оператора SELECT, який перевіряє наявність введених користувачем даних для реєстрації у базі і лише при негативній відповіді зі сторони бази даних відсилається запит за допомогою оператора INSERT, який записує до бази нового користувача.

5.4 Висновок до розділу

У даному розділі було описано структуру розробленого програмного застосунку та приєднаної до нього бази даних, а також надано вичерпну інформацію про компоненти застосунку, їх взаємодію та принцип роботи.

Було розроблено простий інтуїтивно-зрозумілий інтерфейс для програмного застосунку, програмна частина застосунку та база даних, що знаходиться у хмарі, разом з системою взаємодії застосунку та бази даних.

В ході розробки програмного застосунку були використані різні мови програмування та модулі, що створюються за їх допомогою.

					IA61.160БАК.005 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

6 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

6.1 Фази життєвого циклу програмного забезпечення

Розробку програмного забезпечення можна розділити на декілька фаз [18]:

- аналіз вимог;
- проектування;
- розробка і програмування;
- документація;
- тестування;
- реалізація.

Етап аналізу вимог передбачає збір вимог щодо розроблюваного програмного забезпечення, їх систематизацію, документування, аналіз, а також виявлення та розв'язання суперечностей.

На стадії проектування, керуючись вимогами, розробляється високорівневий дизайн системи.

Після того як вимоги та дизайн продукту затверджені, відбувається перехід до наступної стадії життєвого циклу - безпосередньої розробки.

Програмування передбачає чотири основні стадії:

- розробка алгоритмів - фактично, створення логіки роботи програми;
- написання вихідного коду
- компіляція - перетворення в машинний код;
- тестування та налагодження.

Етап документації є доволі умовним та виділяється не завжди. Взагалі умовно беруть чотири рівні документації:

- архітектурна;
- технічна;
- користувацька;
- маркетингова.

					ІА61.160БАК.005 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

6.2 Тестування програмного забезпечення

На основі сценаріїв використання були розроблені тестові випадки та були проведені тести для кожного з них. Тестові випадки та їх результати наведені у таблицях 6.1-6.16.

Таблиця 6.1 – Тестовий випадок «Перевірка виведення інформації про маршрут за замовченням»

Дія	Запустити програмний застосунок.
Очікуваний результат	<ul style="list-style-type: none">– в текстовому полі «Номер маршруту» автоматично вводиться значення маршруту за замовченням;– На елементах виведення виводиться інформація, що стосується маршруту за замовченням.
Результат тесту	Пройшов.

Таблиця 6.2 – Тестовий випадок «Перевірка виведення інформації про обраний користувачем маршрут»

Дія	Ввести в текстове поле «Номер маршруту» номер необхідного маршруту та натиснути кнопку «Оновлення»
Очікуваний результат	<ul style="list-style-type: none">– при початку введення у випадаючому переліку виведуться можливі номери маршруту;– після натискання кнопки «Оновлення» дані на елементах виведення зміняться на ті, що стосуються обраного маршруту.
Результат тесту	Пройшов.

Таблиця 6.3 – Тестовий випадок «Перевірка вибору номеру маршруту, якого немає в базі»

Дія	Ввести в текстове поле «Номер маршруту» значення маршруту, якого немає в базі і натискає кнопку “Оновлення”
Очікуваний результат	<ul style="list-style-type: none"> – при введенні у випадяючому переліку не виводиться номер, який вводить користувач; – після натискання кнопки “Оновлення” елементи виведення стають порожніми.
Результат тесту	Пройшов.

Таблиця 6.4 – Тестовий випадок «Спроба змінити дані, виведені в таблиці»

Дія	Змінити дані, виведені в одній з таблиць.
Очікуваний результат	– Елемент виділяється, але не піддається редагуванню.
Результат тесту	Пройшов.

Таблиця 6.5 – Тестовий випадок «Перевірка виведення даних на діаграмі «Часова характеристика» для певного часового проміжку»

Дія	Ввести в текстові поля «Часова характеристика» діапазон часу, натиснути кнопку «Часова характеристика».
-----	---

Продовження таблиці 6.5

Очікуваний результат	<ul style="list-style-type: none"> – якщо змінилось значення в текстовому полі «Номер маршруту», змінюються дані на всіх інших елементах виведення; – дані, виведені на діаграмі «Часова характеристика», стираються; – на діаграмі «Часова характеристика» виводяться дані, які стосуються обраного проміжку.
Результат тесту	Пройшов.

Таблиця 6.6 – Тестовий випадок «Перевірка виведення даних на діаграмі “Характеристика за зупинками” для певної ділянки маршруту»

Дія	Ввести в текстові поля «Характеристика за зупинками» проміжок маршруту, натиснути кнопку «Характеристика за зупинками».
Очікуваний результат	<ul style="list-style-type: none"> – якщо змінилось значення в текстовому полі «Номер маршруту», змінюються дані на всіх інших елементах виведення; – дані, виведені на діаграмі «Характеристика за зупинками», стираються; – на діаграмі «Характеристика за зупинками» виводяться дані, які стосуються обраного проміжку.
Результат тесту	Пройшов.

Таблиця 6.7 – Тестовий випадок «Перевірка введення невірних даних про проміжок часу»

Дія	Ввести в текстові поля «Часова характеристика» значення часу, які виходять за можливий проміжок часу доби, та натиснути кнопку «Часова характеристика»
Очікуваний результат	<ul style="list-style-type: none"> – якщо змінилось значення в текстовому полі «Номер маршруту», змінюються дані на всіх інших елементах виведення; – дані, виведені на діаграмі «Часова характеристика», стираються; – виводиться повідомлення про помилку.
Результат тесту	Пройшов.

Таблиця 6.8 – Тестовий випадок «Перевірка введення невірних даних про проміжок маршруту»

Дія	Ввести в текстові поля «Характеристика за зупинками» значення часу, які виходять за можливий проміжок маршруту, та натиснути кнопку «Характеристика за зупинками».
Очікуваний результат	<ul style="list-style-type: none"> – якщо змінилось значення в текстовому полі «Номер маршруту», змінюються дані на всіх інших елементах виведення; – дані, виведені на діаграмі «Характеристика за зупинками», стираються; – виводиться повідомлення про помилку.
Результат тесту	Пройшов.

Таблиця 6.9 – Тестовий випадок «Перевірка введення даних про часовий проміжок у неправильному порядку»

Дія	Ввести дані про часовий проміжок у текстові поля «Часова характеристика» в неправильному порядку (менше справа, а більше зліва) та натиснути кнопку «Часова характеристика».
Очікуваний результат	<ul style="list-style-type: none"> – якщо змінилось значення в текстовому полі «Номер маршруту», змінюються дані на всіх інших елементах виведення; – дані, виведені на діаграмі «Часова характеристика», стираються; – виводиться повідомлення про помилку.
Результат тесту	Пройшов.

Таблиця 6.10 – Тестовий випадок «Перевірка введення даних про проміжок маршруту у неправильному порядку»

Дія	Ввести дані про часовий проміжок у текстові поля «Характеристика за зупинками» в неправильному порядку (менше справа, а більше зліва) та натиснути кнопку «Характеристика за зупинками»
Очікуваний результат	<ul style="list-style-type: none"> – якщо змінилось значення в текстовому полі «Номер маршруту», змінюються дані на всіх інших елементах виведення; – дані, виведені на діаграмі «Характеристика за зупинками», стираються; – виводиться повідомлення про помилку.

Продовження таблиці 6.10

Результат тесту	Пройшов.
-----------------	----------

Таблиця 6.11 – Тестовий випадок «Перевірка авторизації користувача»

Дія	Натиснути кнопку «Авторизація», ввести дані користувача и натиснути кнопку «Авторизація».
Очікуваний результат	<ul style="list-style-type: none"> – відкривається вікно авторизації; – перевіряється наявність введених даних із базою даних; – авторизація проходить.
Результат тесту	Пройшов.

Таблиця 6.12 – Тестовий випадок «Перевірка неправильної авторизації»

Дія	Натиснути кнопку «Авторизація», ввести неправильні дані користувача и натиснути кнопку «Авторизація».
Очікуваний результат	<ul style="list-style-type: none"> – відкривається вікно авторизації; – перевіряється наявність введених даних із базою даних; – виводиться повідомлення про помилку; – авторизація не проходить.
Результат тесту	Пройшов.

Таблиця 6.13 – Тестовий випадок «Перевірка реєстрації користувача»

Дія	Натиснути кнопку «Авторизація», натиснути кнопку «Реєстрація», ввести дані нового користувача, натиснути кнопку «Реєстрація».
Очікуваний результат	<ul style="list-style-type: none"> – відкривається вікно авторизації; – відкривається вікно реєстрації; – перевіряється наявність введених даних із базою даних; – новий користувач зареєстрований та доданий до бази даних.
Результат тесту	Пройшов.

Таблиця 6.14 – Тестовий випадок «Перевірка виведення рекомендацій по оптимізації розкладу руху транспортних засобів на обраному маршруті»

Дія	Натиснути кнопку «Рекомендації».
Очікуваний результат	<ul style="list-style-type: none"> – Проводяться розрахунки по оптимізації розкладу руху транспортних засобів; – Відкривається вікно рекомендацій; – Виводяться рекомендації по оптимізації розкладу руху транспортних засобів;
Результат тесту	Пройшов.

Таблиця 6.15 – Тестовий випадок «Перевірка оновлення даних»

Дія	Оновити базу даних та перезапустити програмний застосунок.
Очікуваний результат	– в елементах виведення виводяться дані, що стосуються оновленої інформації з бази даних.
Результат тесту	Пройшов.

6.5 Висновок до розділу

Тестування програмного забезпечення – один з найважливіших і найбільш трудомістких етапів розробки програмного забезпечення, який допомагає зрозуміти, чи коректно працює програмний застосунок, а також виявити можливі помилки та неточності при його роботі. Проводиться у декілька етапів та супроводжує весь процес розробки.

Тестування програмного застосунку показало, що всі модулі є справними та працюють саме так, як і повинні. Всі можливі помилки, які час від часу трапляються з програмним застосунком, були завчасно усунуті, а випадки, коли вони могли статися, перериваються із виведенням повідомлення про помилку, хоча в іншому випадку вони могли призвести до некоректної роботи застосунку чи навіть аварійного переривання її роботи.

7 ВПРОВАДЖЕННЯ ТА ВИКОРИСТАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

7.1 Апаратні та програмні вимоги для експлуатації програми

Розроблена система створена за допомогою мови програмування C#, тому апаратні та програмні вимоги до сервера, на якому буде експлуатуватись співпадають з вимогами C#.

Мінімальні системні вимоги до сервера описані у таблиці 7.1, а рекомендовані – у таблиці 7.2.

Таблиця 7.1 – Мінімальні системні вимоги до апаратної частини сервера

Процесор	2-ядерний процесор з тактовою частотою 1.3 ГГц або більше.
Місце на диску	1 гігабайти.
Оперативна пам'ять	1 гігабайти.
Операційна система	Windows 7 або пізніша

Таблиця 7.2 – Рекомендовані системні вимоги до апаратної частини сервера

Процесор	2-ядерний процесор з тактовою частотою 2.6 ГГц або більше.
Місце на диску	1 гігабайт.
Оперативна пам'ять	2гігабайти.
Операційна система	Windows 7 або пізніша.

Програмний застосунок не потребує ніяких додаткових програмних засобів для своєї роботи та не потребує встановлення.

Для коректної роботи програмного застосунку необхідно підключення до мережі Інтернет.

7.2 Інструкція користувача

При запуску програми відповідно до базових налаштувань в ній одразу виводиться інформація по маршруту, який є першим в списку маршрутів, що знаходиться в базі. При цьому діаграми виводять чисельну інформацію щодо повної протяжності маршруту протягом всього дня. При натисканні кнопки «Оновлення» інформація на діаграмах не змінюється, тому що значення текстових полів не змінюється.

Текстове поле «Номер маршруту» має функцію автозаповнення. Тому, коли починається введення в це поле певного значення, у випадаючому меню виводяться усі доступні в базі номери маршрутів, що починаються з введеного числа. При натисканні лівою кнопкою миші на один з випадаючих номерів поле автоматично заповниться обраним значенням.

Таблиця «Завантаженість» відображає загальні характеристики місткості кожної з присутніх на маршруті моделей автобусів та залежну від неї характеристику «коефіцієнт навантаження». Цей коефіцієнт є відношенням середньої за день кількості пасажирів, що були перевезені автобусами цієї моделі, та їх оптимальною завантаженістю. Крім того клітини з цією інформацією мають інтуїтивно зрозумілу кольорову індикацію: якщо коефіцієнт навантаження нижче або дорівнює одиниці, то клітина буде зеленою, а якщо вище одиниці, то червоною.

В таблиці “Загальна характеристика” виводяться всі присутні в базі даних автобусні маршрути. В текстове поле «Номер маршруту» можна вводити лише присутні у першому стовпчику «Номер маршруту» номери, щоб отримати інформацію з інших елементів інтерфейсу. При введенні номерів, яких немає в таблиці, програма виведе порожні значення таблиці «Завантаженість» та діаграм.

					IA61.160БАК.005 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Для виведення необхідної інформації на діаграмах необхідно вводити в текстові поля під ними лише ті значення, що входять в діапазон значень. Для часу це - цілі значення від 0 до 24, для ділянок маршруту - номери зупинок, зображених на карті, починаючи з 1. При введенні не цілих значень програма проігнорує частину числа після коми. Менше значення (нижню границю діапазону) необхідно вводити в ліве текстове поле, більше значення (верхню границю) - в праве поле.

При введенні значень, що не входять в діапазон допустимих значень, або при некоректному введенні більшого, меншого чи обох значень в поля введення, буде виведено повідомлення про помилку.

Загальні рекомендації по оптимізації обраного маршруту, розраховані програмою за алгоритмом оптимізації, виводяться в окремому вікні при натисканні кнопки «Рекомендації». Рекомендації виводяться у вигляді незмінюваного тексту.

7.3 Висновок до розділу

Визначення системних та програмних вимог – важлива частина розробки програмного забезпечення. Правильно складена документація щодо цих вимог може зберегти велику кількість часу розробників та користувачів, які будуть працювати з розробленого програмного застосунку. Також важливим є складання інструкції користувачу розробленого програмного застосунку, щоб надати можливість використовувати програмне забезпечення пересічним користувачам комп'ютерів, а не тільки спеціалістам з інформаційних технологій. Інструкція не має бути складною та містити термінологію, що може бути незнайома користувачеві.

					IA61.160БАК.005 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

При виконанні даного дипломного проекту було створено програмний застосунок для статистичного аналізу пасажиропотоку громадського транспорту.

На першому етапі було проведено аналіз предметної області розроблюваного програмного застосунку. Було визначено цілі та задачі розробки, а також висунуто основні вимоги до застосунку. Аналіз існуючих рішень дав змогу зробити висновок, що прямих аналогів розроблений програмний застосунок не має. На основі отриманих даних були розроблені функціональні та нефункціональні вимоги до програмного застосунку. Було проведено вибір технологій розробки, з обґрунтуванням цього вибору. Далі, на основі вимог, визначених раніше, за допомогою обраних технологій було створено програмний застосунок. На наступному етапі були відтворені тестові випадки, та на їх основі проведено тестування роботоздатності програмного застосунку. У заключній частині було описано апаратні та програмні вимоги, та вказано інструкцію для користувача.

Розроблений програмний застосунок є самостійним завершеним продуктом, що має потенціал для розширення та інтегрування в більші системи моніторингу та керування міським транспортом. Він має великі модернізаційні можливості за рахунок приєднання до нього вже існуючих систем, таких як онлайн карти чи системи спостереження за рухом транспортних засобів на основі технології GPS. Крім того, структура програмного застосунку є достатньо гнучкою для можливості подальшої розробки та створення нових модулів на основі вже існуючих.

У ході розробки програмного застосунку було ознайомлено з новими технологіями та покращено навички роботи із раніше відомими. Було отримано досвід роботи з хмарними системами та взаємодії між створеним додатком та базою даних, що знаходиться в хмарі.

					IA61.160БАК.005 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Client-server architecture | computer science | Britannica. URL: <https://www.britannica.com/technology/client-server-architecture> (дата звернення: 15.05.2020).
2. What Is Cloud Computing. URL: <https://www.salesforce.com/products/platform/best-practices/cloud-computing/> (дата звернення: 16.05.2020).
3. О.Г. Осауленко. Статистичний словник. Київ: 2012. 500 с.
4. Валідатор. URL: <https://uk.wikipedia.org/wiki/Валідатор> (дата звернення: 5.05.2020).
5. Jörg Eberspächer, Hans-Joerg Vögel, Christian Bettstetter, Christian Hartmann. GSM - Architecture, Protocols and Services. John Wiley & Sons, 2008. 338 с.
6. Леоненков. Самоучитель UML. URL: <http://khp-iip.mipk.kharkiv.edu/library/case/leon/gl4/gl4.html> (дата звернення: 31.05.2020).
7. Введение в C#. Язык C# и платформа .NET Core. URL: <https://metanit.com/sharp/tutorial/1.1.php> (дата звернення: 01.06.2020).
8. Общие сведения о Visual Studio | Microsoft Docs. URL: <https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2019> (дата звернення: 01.06.2020).
9. Windows Forms overview | Microsoft Docs. URL: <https://docs.microsoft.com/en-us/dotnet/framework/winforms/windows-forms-overview> (дата звернення: 01.06.2020).
10. Обзор конструктора Windows Forms | Microsoft Docs. URL: <https://docs.microsoft.com/ru-ru/visualstudio/designers/windows-forms-designer-overview?view=vs-2019> (дата звернення: 01.06.2020).
11. SQL - Энциклопедия языков программирования. URL: <http://progopedia.ru/language/sql/> (дата звернення: 01.06.2020).

					IA61.160БАК.005 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

12. Что такое PostgreSQL. URL:
http://www.sai.msu.su/~megera/postgres/talks/what_is_postgresql.html (дата
 звернения: 01.06.2020).

13. PostgreSQL Clients. URL:
https://wiki.postgresql.org/wiki/PostgreSQL_Clients#pgAdmin_4 (дата звернения:
 01.06.2020).

14. Использование инструментов Visual Studio для создания базы данных.
 URL: <https://it.wikireading.ru/26023> (дата звернения: 01.06.2020).

15. Chris Dawson, Ben Straub. Building Tools with GitHub: Customize Your
 Workflow – O'Reilly Media, 2016 – 302 pages

16. Что такое Azure — облачные службы от Майкрософт. URL:
<https://azure.microsoft.com/ru-ru/overview/what-is-azure/> (дата звернения:
 05.06.2020)

17. Паттерны проектирования в C# и .NET. URL:
<https://metanit.com/sharp/patterns/> (дата звернения: 5.05.2020).

18. Стадии цикла разработки ПО - QALight. URL:
<https://qalight.com.ua/baza-znaniy/stadii-tsikla-razrabotki-po/> (дата звернения:
 5.06.2020).

					IA61.160БАК.005 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		